```
************************************************************
                                  ***
        /////////      ////   ////  ***  /////////
       IIIIIIIII/     IIII/  IIII/    ***IIIIIIII//
      IIIII/          IIII/  IIII/     IIII/  IIII/
      IIIII/          IIII/  IIII/     IIII/* IIII/
      IIIII/          IIII/  IIII/     IIII/***III/
      IIIII//////  // IIII///IIII/  // IIII///***I/  //
       IIIIIIIII/ II/  IIIIIIIII/  II/  IIIIIIII*** II/
                                                ***
************************************************************
```

CIRCULO DE USUARIOS DE QL

****

Fanzine mensual independiente para usuarios de Sinclair QL y compatibles

AÑO 2 NUMERO 10 JULIO DE 1989

Estamos en el año 9 Después de Sinclair. Toda Hispania está ocupada por las legiones de PCs. ¿Toda? ¡No! Un puñado de irreductibles QLs resiste todavía y siempre ante el primitivo invasor...

CONTENIDO

INFORMACION SOBRE C.U.Q.

Compilación de colaboraciones: Salvador Merino
Títulos y texto "Información sobre C.U.Q.": Marcos Cruz
Sub-Distribución: Marcos Cruz
Imagen de la portada: THE SOLUTION

El fanzine C.U.Q. es un medio de comunicación e intercambio de ideas y conocimientos entre usuarios de ordenadores Sinclair QL y compatibles. El fanzine se realiza mediante la colaboración desinteresada de todas las personas que lo desean.

¿COMO SE PUEDE RECIBIR EL FANZINE?

El fanzine se envía a quien contribuye a su realización aportando alguna colaboración, o bien a quien lo solicita sin más.

Quienes sólo quieran recibir el fanzine, sin enviar ninguna colaboración

para que sea incluida en él, deberán enviar UN disco formateado de 3.5 pulgadas o DOS microdrives formateados, junto con un sobre franquedo y con las señas escritas para el reenvío. Estos discos o microdrives SIN COLABORACIONES deben enviarse a: Marcos Cruz, Acacias 44, 28023 Madrid

El fanzine de cada mes estará disponible, para ser enviado a quien lo solicite, a finales de la primera semana de dicho mes, con lo cual pueden enviarse las peticiones a comienzos de mes.

¿COMO SE PUEDE COLABORAR EN LA REALIZACION DEL FANZINE?

Quien desee colaborar en la realización del fanzine, debe igualmente enviar UN disco de 3.5 pulgadas o DOS cartuchos de microdrive (aunque sólo uno contenga algo) junto con un sobre franquedao y con la dirección escrita para el reenvío. Estos discos o microdrives CON COLABORACIONES deben enviarse a: Salvador Merino, Ctra. Cádiz (Cerámicas Mary), 29640 FUENGIROLA (Torreblanca del Sol), MALAGA

¿Qué puede enviarse como colaboración? Cualquier cosa: artículos, cartas, opiniones, preguntas, respuestas, ideas, sugerencias... Si bien el nexo de unión entre quienes hacen posible C.U.Q. es el interés y la inquietud por el ordenador Sinclair QL, y por la informática en general, y esa es la temática global del contenido del fanzine, C.U.Q. acepta artículos sobre cualquier tema que pueda interesar a los lectores.

Con todas las colaboraciones recibidas a lo largo de un mes, se confecciona el fanzine correspondiente al mes siguiente. Los discos y microdrives con los boletines se echarán al correo en el último fin de semana del mes.

No deben enviarse más colaboraciones hasta haber recibido de vuelta los discos o microdrives con el fanzine del mes correspondiente.

FORMATO DE LAS COLABORACIONES

Los textos deben ser ficheros _doc del programa Quill, con los siguientes márgenes: izquierdo=1, derecho=80, sangrado=5. Los textos deben ir justificados a la derecha, y no deben emplearse caracteres no castellanos cuyo código ASCII sea mayor de 127. Tampoco deben usarse los corchetes cuadrados, ni las llaves, ni las barras vertical o inclinada hacia la izquierda.

Los artículos deben agruparse temáticamente según las siguientes secciones:

CAR	Cartas abiertas
NOT	Noticias
LIB	Comentarios de libros y otras publicaciones
PRO	Comentarios de programas
HAR	Comentarios sobre hardware, y montajes
BAS	Lenguaje SuperBASIC
FTH	Lenguaje Forth
LEC	Lenguaje C
ASM	Lenguaje ensamblador 68000
COB	Lenguaje COBOL
MAN	Manuales traducidos
ZET	Z88
SPE	Spectrum
OFE	Ofertas, compra-venta, anuncios particulares
OTR	Otros temas de interés, aunque no sean informáticos.

Esta lista podrá ser ampliada o modificada por el recopilador, a sugerencia de los colaboradores.

En el caso de que la colaboración consista en preguntas, o respuestas, deberá igualmente incluirse en una de las secciones indicadas, según el tema de a que se refieran.

Todas los artículos que pertenezcan a una misma sección deben ir en un mismo fichero. El nombre de este fichero tendrá el formato siguiente: SSSxxnnn_doc, donde "SSS" son los tres caracteres correspondientes a la sección de destino del texto, "xx" es el número de fanzine al que se envía el texto y "nnn" son las iniciales del autor.

En el texto, el título de cada artículo deberá ir en mayúsculas, en letra negrita, y a la izquierda de la línea (en el margen de sangrado). Excepto la letra negrita en los títulos de los artículos, no deberá emplearse ningún tipo de letra (subrayada, negrita, subíndice o superíndice) en los textos.

Deberá dejarse siempre UNA SOLA línea en blanco para separar el título de un artículo del artículo anterior, si lo hay. Dentro de cada artículo deberá evitarse por completo el incluir líneas en blanco, tanto para separar párrafos como por cualquier otro motivo, con el objeto de ahorrar espacio en las copias impresas.

Al final de cada cada artículo debe ir el nombre del autor, en la columna 40, seguido en la misma línea por la localidad y la fecha.

Algunas importantes recomendaciones a observar:

-Déjese siempre un espacio detrás de cada signo de puntuación (coma, punto, punto y coma, dos puntos) ¡pero no delante!

-No deben aislarse con espacios los signos de interrogación, admiración, guiones o paréntesis, sino que deben estar unidos a la palabra siguiente -si son signos de apertura- o anterior -sin son de cierre.

-Cuídense tanto la ortografía y las tildes como los errores de "tecleo". Para ello, es recomendable releer lo que se ha escrito.

¿EN QUE FORMATO SE RECIBE EL FANZINE?

El fanzine se recibe en forma de varios ficheros _doc del procesador Quill, al objeto de permitir el acceso al mísmo de los usuarios que no tienen unidades de disco y/o ampliación de memoria.

El nombre de cada fichero indica el número de fanzine al que pertenece y la sección que contiene.

Para componer el fanzine debe cargarse el fichero CUQxxPOR (donde "xx" es el número de fanzine) y seguidamente deben UNIRse al final del texto, uno a uno, los ficheros de las restantes secciones en el orden que indique la columna "Sección" del índice del fanzine. Seguidamente puede alterarse el número de líneas por página, asignarse un pie de página y/o un encabezamiento y, si se desea, añadir cambios de página forzados para evitar que los títulos de secciones o artículos queden al final de alguna página. Finalmente, se ha de grabar el fanzine completo en disco y/o sacar una copia por impresora.

Si, por falta de memoria, no puede componerse todo el fanzine entero, deberá imprimirse por partes, a juicio de cada uno.

Una vez tenemos el fanzine en papel, habrá que rellenar a mano los números de página del índice.

¿PUEDEN INCLUIRSE DIBUJOS EN EL FAZINE?

Normalmente la portada llevará un dibujo (preferiblemente relacionado con alguno de los artículos del fanzine). El dibujo será un fichero de pantalla, en modo 4 u 8. El nombre de estos ficheros debe ser: DBPxxnnn_MODEm, donde "m" es el modo de pantalla en el que están realizados. Como en los textos, "xx" es el número del fanzine de destino, y "nnn" son las iniciales del autor.

El dibujo elegido por el recopilador para ser la portada del mes, se incluirá en el fanzine con el nombre CUQxxDBP_MODEm (donde "xx" es el número de fanzine y "m" indica el modo de pantalla, 4 u 8). Este fichero deberá emplearse para sacar un "hard-copy" sobre el folio de la portada.

La inclusión de dibujos en los artículos aún no es posible, pero están en estudio varios métodos posibles.

¿COMO ENVIAR PROGRAMAS AL FANZINE?

El fanzine C.U.Q. sirve también como intercambio de programas (PROGRAMAS REALIZADOS POR LOS MISMOS COLABORADORES, NO PROGRAMAS COMERCIALES). Estos programas pueden estar realizados en cualquier lenguaje disponible para el QL. En la medida de lo posible, y siempre que su longitud no sea excesiva, los textos fuente de dichos programas deben incluirse en un artículo (en la sección del lenguaje correspondiente) en el que se explique el funcionamiento u otros detalles que el autor considere oportuno. Los listados de programas que se incluyan en el boletín, no deberán tener líneas con más de 80 caracteres. Si el autor prefiere no ceder al fanzine el texto fuente, deberá igualmente escribir las instrucciones de manejo en un pequeño artículo, para su inclusión en el fanzine.

El fanzine C.U.Q. no mantiene una librería de programas como tal, puesto que, por enviarse éstos en el mismo disco del fanzine, no es necesaria. No obstante, si alguien precisa alguno de los programas, puede solicitarlo.

Normas a tener en cuenta para enviar programas al fanzine.

-Póngase a cada fichero una extensión adecuada que indique su tipo, por ejemplo:

_bas si es un programa en SuperBASIC
_exe o _task si es un programa EXECutable
_bin si es código máquina no EXECutable
_txt o _asc si es un fichero de código ASCII
_fth si es un texto fuente es Forth
_scr si es una pantalla, o un fichero fuente del Forth de Computer One
...etcétera

-Póngase a todos los ficheros de un mismo programa un mismo nombre inicial, que no coincida con alguna de las referencias de los programas ya existentes. Si entre los ficheros de un programa existe un fichero _doc del Quill, o de otro paquete de Psion, el nombre inicial común a todos los ficheros no podrá, por tanto, ser más largo de 8 caracteres.

-No deben emplearse en los nombres de los ficheros caracteres que no sean aceptados directamente, es decir, que no sea necesario encerrar el nombre entre comillas para manipular el fichero.

-Deben escribirse instrucciones detalladas del manejo de los programas, bien dentro del mismo programa, o en un artículo que aparezca en el fanzine, o bien en un fichero _doc.

Los programas deben enviarse, como las colaboraciones escritas, a: Salvador Merino, Carretera de Cádiz (Cerámicas Mary), 29640 FUENGIROLA (Torreblanca del Sol), Málaga.

EDITORIAL

Prácticamente este nuevo número se está preparando, como siempre, a principio de este mes de JUNIO para que esté listo para copiar el dia 25 de JUNIO.

Como ya me estaba temiendo, en verano la gente se duerme. Algunos por los examenes finales, otros por las vacaciones, y algunos como yo, nos toca trabajar todavía más que en invierno.

Tenemos un par de miembros fuera de servicio, porque haciendo algunas modificaciones con el soldador y sacando/metiendo tarjetas han averiado accidentalmente sus QLs. Espero que sea temporal, porque su labor en el pasado ha sido bastante buena, ¡Ya veremos!.

Y hablando de mi trabajo, prácticamente se podria decir que mi trabajo es mayormente dirigir una empresa, pero aunque muchos trabajitos son hechos por gente que trabajan para mi, existen algunos trabajos de oficina, almacén, cajero, etc..., los cuales se podrian calificar de privados (no me gusta que ninguno de mis empleados conozca al detalle algunos aspectos del negocio).

Para controlar alguno de mis cuatro negocios familiares (¡Ojo!, yo solamente puedo gastar mi asignación semanal), se podria decir que el ordenador no sirve para casi nada. Solamente para escribir cartas, bases de datos de direcciones, hacer gráficos tipo EASEL por entretenimiento, y no sé que más. Por lo que a mi respecta si no fuese tan malo escribiendo con la máquina de escribir mecánica, podriamos decir que el ordenador no me seria útil en absoluto.

No hay que ser adivino para saber que mis comienzos fueron con los video juegos Atari, y más tarde, despué s de desviarme con el vicio de piratear juegos con el Spectrum, con el QL me metí en el otro vicio de aprender a programar y leerme todos los libros y revistas relacionadas con el tema.

En verano no trabajo más horas, simplemente es la mejor época del año para salir y tener relaciones con amigos (ir a la playa, de copas, al billar, etc..). Pero este año quizás me lleve el Z88 a la playa y a los bares (la única vez que lo he hecho, se me echaron encima una nube de curiosos). EL UNICO PROBLEMA ES QUE SECTOR SOFTWARE SE ESTA RETRASANDO EN ENVIARME EL CABLE & SOFT.

Espero haber dejado claro, que no hay que preocuparse, voy a escribir todo lo que pueda (si solamente me dedicase a traducir el material en Inglés, habria CUQ para más de 15 años), y grabar CUQ a los miembros actuales, es una tarea que si se hace desde RAMDISK en cadena todo junto, no llega ni a los 20 minutos. Y eso no es mucho tiempo, ¡Verdad!.

Tengo que agradecer a todos aquellos que me han comunicado que hacen copias de CUQ a sus conocidos, porque necesitamos que CUQ llegue al mayor número de usuarios de QL sin que yo o Marcos (único distribuidor voluntario para cuando no se envian colaboraciones) tengamos que malgastar nuestro valioso tiempo, y asi podamos dedicarnos a otras cosillas más interesantes para todos los lectores.

Las noticias sobre el QL son bastante buenas (ver sección noticias).

A última hora, son ya tres los miembros que se han cargado su QL accidentalemente. A este paso nos quedamos sin miembros de ELITE. Necesitamos más miembros que sean capaces de aportar algo.

Con el CUQ 10 se llena el tercer disco.

S. Merino

NOTICIAS

LAS NOTICIAS MAS INTERESANTES DE PRIMAVERA

La revista SINCLAIR QL WORLD ha dejado definitivamente de distribuirse en los kioskos y librerias. Pero eso no quiere decir que nos hemos quedado sin QL WORLD, simplemente que si queremos seguir obteniendola, debemos subscribirnos.

Yo me he subscrito hace poco a QL WORLD desde el número de abril por correo aéreo, y he recibido los números de abril y mayo 2 semanas después.

La revista QL WORLD no ha perdido calidad, ni mucho menos con el cambio, incluso diria que en estos dos números ha aumentado el número de páginas dedicadas a los listados en ASSEMBLER o SUPERBASIC.

Existen muchas revistas solamente disponibles para subscriptores, debido a que solamente interesan a un público muy minoritario. Por lo tanto, no os preocupeis mientras exista un número medio decente de usuarios QL interesados por conocer o adquirir todas las novedades en hardware y software para el Legendario Sinclair QL.

Si de SOLUTION era suficiente para algunos, pero pecaba de lento, existe un nuevo producto que según SCHON (el posible distribuidor en UK), no se le puede llamar PC Emulator. Todo hace pensar que se trata de una tarjeta que se conecta al port-ROM del QL, el software en ROM, CGA implementada por Hardware,... Si como se supone, este producto llamado TRANSFORMER es capaz de conseguir la velocidad de un PC XT, si no sale otra cosa mejor, se puede convertir fácilmente en la tarjeta hardware más vendida en el mundo QL (siempre y cuando sea muchisimo más barato que un PC).

El IBM PC EMULATOR II de Ant Computing se ha convertido en desaparecido en comabate II, y no me extrañaria nada que THE SOLUTION corriese la misma suerte en los próximos meses, si el TRANSFORMER es lo que parece ser.

El último invento de Tio Clive Sinclair es la SINCLAIR SATELLITE, que consiste en un satellite TV system por 195 libras con 48 canales, stereo,...

El THOR XVI sigue su carrera particular a paso tortuga (¡Qué a lo tonto van ya por 3 años!) con la promesa de que en Abril la nueva compañia podrá aumentar la producción de cientos de máquinas a miles. Por lo pronto los nuevos precios sin impuestos son: THOR XVI con un sólo disco 1.080 libras, con un disco y disco de 20 Megas 1.550 libras, con dos discos y disco 45 Megas 1.865 libras y cada 512K de RAM adicional cuesta 155 libras. A mi me parece demasiado caro, pero po lo que se ve y se oye, hay empresas que lo están comprando en Dinamarca por su sistema operativo ARGOS y principalmente su NETWORK. Otra ventaja es poder trabajar en varios lenguajes (Inglés, Ruso, Alemán,...) en multitarea. Yo lo único que sé es que no puedo comprarlo actualmente y a ese precio deberia incluso que pensarmelo dos veces como minimo (si por lo menos tuviese un MC 68020 en vez de un MC 68000).

1989 en el mundo del QL se podria llamar el año del disco duro, han salido más de 6 interfaces de disco duro nuevos este año + los viejos, pero el de Miracle Systems es el más barato, y quizás el más lento.

El programa BASIC-port ha sido actualizado por SUPERBASIC-PORT (PDQL) que es totalmente compatible ANSI y LATTICE, y es capaz de traducir un programa en Superbasic a C, listo para compilar en un IBM PC.

La revista ORDENADOR PERSONAL (la primera revista Española de ordenadores personales) ha cambiado mucho desde su nacimiento. Antes era interesante, pero ahora se ha convertido en una más del montón de revistas dedicadas exclusivamente a los PC-compatibles. Sus páginas están llenas de articulos que no poseen ningún interés didactico. No me extraña nada el bajo nivel de los usuarios de PCs, solamente son usuarios de programas y consumidores de nuevos modelos de PCs.

Según la revista Ordenador Personal num. 78 Mayo 1.989, los analistas de mercado afirman que en 1991 el 60% del mercado para ordenadores personales será para los PCs, un 25% para PS/2, un 10% para máquinas Macintosh y el 5% para otros posibles fabricantes.

Es totalmente posible que los PCs en 1991 todavía posean la mayor parte del mercado en ventas, pero quizás la mayor parte de las ventas sean equipos usados (hoy en dia el 5% de las ventas totales, y quizás más de un 10% de unidades vendidas). Pero lo que no me creo es que las ventas de Amiga, Atari ST, THOR, NEXT, ATARI ABAQ, ACORN ARCHIMEDES, (todas las máquinas basadas en un 68000), (posibles nuevos ordenadores basados en el MOTOROLA 78000),..., etc, (nuevos modelos en desarrollo)...etc,.. sean solamente el 5% del mercado en ventas. O los analistas están chiflados, comprados, o es la vieja propaganda comecocos que tanto éxito ha tenido estos últimos 3 años (los nuevos PC a base de 80386 solamente se benefician en un aumento de velocidad en MS-DOS, y nada más).

La revista ATARI USER, una de las pocas en Español dedicadas a un 68000, es la revista más pobre que nos podemos echar a la cara. El nivel de conocimientos de sus escritores o colaboradores es tan bajo que parece que no han usado nunca un ordenador o son usuarios de PCs. La tónica de la revista es casi idéntica a

una revista dedicada a los PCs.

La revista MICROHOBBY, la única para Spectrum que se ve por aqui, aunque ahora es mensual, es una de las pocas revistas que podriamos decir que vale la pena. Sus listados de programas, especialmente el AULA SPECTRUM, son muy interesantes para cualquier usuario que desee aprender a programar. Ultimamente por solamente 375 ptas te regalan (o mejor dicho, incluyen en el precio) una cinta que contiene un juego antiguo comercial original (normalmente DINAMIC), un juego en código máquina hecho por un lector, y algunos cargadores especiales para los juegos comerciales comentados. Solamente por la cinta, ya vale la pena comprar Microhobby (es una ganga).

Ultimamente se habla mucho de un fenómeno llamado VIRUS que solamente se da en los PCs. También se habla de VACUNAS contra esos virus, las cuales son temporales o definitivas. Aunque el lector está superconfundido, esos virus en nuestro QL se llaman BUGs. Lo que significa que son programas que contienen errores que luego alteran ficheros debido a que el MS-DOS siempre está haciendo llamadas al disco (p.e.: el disco duro).

En un QL cuando existe un BUG, solamente puede ocurrir dos cosas: el programa se cuelga (si es en multitarea solamente el programa en cuestión), o continua sin hacer lo previsto hasta que se cuelga cuando menos te lo esperas. Un ejemplo de vacuna podria ser una rutina para apaliar el BUG MG de la rutina POINT que imprime dos puntos en vez de uno.

En un Z88 un BUG puede estar oculto sin ser detectado durante semanas e incluso meses, mandando el Z88 a la porra en el momento menos esperado.

S. MERINO, Fuengirola, 1/6/89.

THOR XVI VERSION 4

Se suministra con el sistema operativo ARGOS V6.40/1.07, que introduce nuevas facilidades para el manejo del SCSI hard disc. ARGOS 6.40 ahora soporta teclados IBM AT Standard, IBM AT extendidos y PS/2 extendidos, cubriendo 11 lenguajes nacionales diferentes.

Ahora es posible usar las avanzadas facilidades de cualquier impresora, incluyendo laserprinters, debido a las nuevas tablas traductoras definidas por el usuario.

ARGOS soporta una separada screen MODE para cada job (igual que QRAM).

Además de PSION Xchange, se suministra 2 discos con un sistema de menu fácil de usar y utilidades.

Existe una lista precios de 12 páginas de nuevos productos para el CST THOR XVI, conteniendo también descripción de nuevos productos en preparación.

También se anuncia un nuevo editor para programadores, ARCTURUS EDITOR, y un nuevo compilador C, C++ COMPILER. Ambos corren en THOR y QL.

Más información:
THOR INTERNATIONAL COMPUTER SYSTEMS I/S
Raadhusatraede 4 b, 4.sal, DK 1466 Copenhagen K
Mail to: P.O. Box 59, DK 1002 Copenhagen K, Denmark.
Phone no: +45 1 930305 (after May 15th: +45 33930305)
Fax no: +45 1 938292 (after May 15th: +45 33 938292)

TRANSFORMER

He recibido el QL WORLD de Junio, y SCHON desvela timidamente algunos detalles de su nuevo producto.

Transformer es el emulador PC/MS-DOS más rápido para el QL. El prototipo consiste en puramente en una tarjeta hardware que se conecta al port de expansión. Está operando al doble de velocidad que los productos de la competencia (si se refiere al THE SOLUTION estamos arreglados). Todo el software es suministrado en ROM y CGA basada en hardware. Transformers es suministrado con un duplicado del port de esxpansión. Desarrollado fuera de UK y construido en UK.

No es por nada, pero si el TRANSFORMER solamente corre el software MS-DOS el doble de rápido que THE SOLUTION, me quedo con THE SOLUTION (en un ATARI ST/QL Emulator o THOR XVI corre 2.5 veces más rápido). Aunque CGA basada en hardware significa poder correr más programas. Tampoco está muy claro si Transformer se puede instalar con la TRUMP CARD (768K), pues no hay espacio ROM para un emulador de PC, y si solamente se puede usar con expansiones de 512K RAM, o menos, nuestros QLs van a transformarse de portaviones a superpetroleros. Lo más lógico es esperar a los comentarios sobre el TRANSFORMER, y un consejo, si necesitais trabajar en serio con programas MS-DOS, comprar un IBM PC XT o AT de segunda mano (un original es siempre un original), o un PC compatible barato, no seria mala idea. Eso si, el MS-DOS para hacer lo que a mi me gusta es

horrible, prefiero el QDOS.

LOS MICRODRIVES AL BORDE DE LA EXTINCION

Se sabia que podria ocurrir, pero no se esperaba hasta después de los 8 años prometidos (1994).
La compañia Ablex va a dejar de fabricar los MDVs este año. La razón es un declive de las ventas.
Hoy en dia casi la mayoria de los usuarios adictos al QL usan discos de 3.5", algunos usan de 5.25" como alternativa más barata o para usar discos en formato MS-DOS, y muy pocos usan discos 3".
1989 es el año del disco duro, y más de uno va a ampliar su QL con alguno de los más de seis interfaces de disco duro disponibles y con cualquier unidad de disco duro que pueda permitirse su bolsillo.
Todavía tengo un QL de repuesto, aún no me he decidido si voy a ampliarlo a una configuración diferente a la que poseo en el QL actual, o simplemente esperar un CST THOR, pues estoy seguro que un año de estos voy a comprar uno de ellos (solamente hay que echar paciencia al asunto).

S. Merino, Fuengirola, 7/6/1989.

CARTAS ABIERTAS

UN ALTO GRADO DE IGNORANCIA

A raíz de una dura reprimenda de Salvador, echándome en cara "un alto grado de ignorancia", en las cosas de "nuestro QL", (por no saber ver la portada de CUQ), quiero hacer una llamada de atención sobre lo siguiente:
Las personas que he conocido con un ordenador, siempre han tenido un grave problema que es muy sencillo: no entender lo que se lee (bien por no saber inglés, o bien por la dificultad del tema), y yo soy uno de estos.
El no disponer de suficiente tiempo y la "ineducación informática" con la que empecé con el QL, me impide hablar (entender) de tú a tú con nuestros compañeros.
Problemas tan sencillos como utilizar un GraphiQL, Trumpcard, Toolkit, o Qram, se convierten en insolubles en ocasiones por culpa del inglés. Volcar una pantalla desde Qram, ha sido siempre cosa de churro, y desde luego, con el manual de Trumpcard, la cosa no ha mejorado.
Reconozco que esto se debe a mi entusiasmo por la programación en Archive, a lo que me he dedicado prácticamente en exclusiva, y al que todavía no he sacado todo su jugo. Incultos como yo los había a montones en Qlave y de verdad lamento que en CUQ no los haya.
Mi último problema con Archive (a ver si alguien me lo puede solucionar), es el siguiente: Al programar una línea Lprint, ésta admite cualquier Tab (60, 100, 115, etc), pero al imprimir se ríe de los Tab y a partir del 80, mezcla todos los siguientes. Aclaro que mi impresora es una BMC e imprime con letra comprimida con lo que deberían salir 132 columnas (creo). Al hacer un Dump de un fichero Archive, sí que respeta 132 columnas. Me cojo la instalación de impresora y veo que figura con 80 columnas; lo modifico a 132 y seguimos igual. Así como en Abacus la opción Design permite fijar el ancho al imprimir, en Archive no veo cómo hacerlo. Si podéis solucionarme este lío tendré una pega menos en el futuro. Me he leído los Qlave varias veces y los CUQ por lo menos dos veces pero hay cosas que siempre quedan oscuras. En el libro "Archive" de Anaya tampoco se aclara nada sobre esto. En Abacus y Archive ya sé cómo meter letra comprimida y ampliada (los chr 15 y 14) pero no sé cómo anularlo. Si meto después de chr(15) un chr(14) se produce otro tipo de letra.
Espero que mi incapacidad informática no me impida seguir recibiendo este estupendo medio de información y formación que es CUQ.
Hasta otra,

Josu Regidor Eguren Lezama 15-6-89

COMENTARIOS DE PROGRAMAS

¡Y SEGUIMOS CON THE SOLUTION...............!

¡Ya está!¡Por fin lo tengo!.Me acaba de llegar el programa estrella de los últimos tiempos dentro del software de nuestro QL.Como ya tengo referencias de él por los boletines cuq7 y cuq8 me siento mediatizado en cuanto a la opinión que hubiere de tener respecto a que,haciendo bueno el nombrecito,sea realmente la "solution" de que hace gala Digital.

Queramos o no,estamos rodeados de PCs,con una coleccion de software apabullante en cuanto a calidad y cantidad y con unas dotaciones de hardware(tarjetas,discos,discos duros,raton,etc)que hacen que en el fondo no nos importaria tener un conjunto de esos sin tener que por ello renunciar a nada de lo que ya tenemos.

Quiero decir que el emulado siempre hace mejor las cosas que el emulador y además mas rápido.Si el programita se llama como se llama es porque pretende solucionar un problema latente.Si el personal lo espera como agua de mayo es porque en el fondo queremos,sin renunciar a nuestro QL,meter la nariz en otro mundo paralelo al nuestro pero distante.Y eso precisamente es lo que el programita hace.Para mi,despues de torearlo arriba y abajo,probar con él un montón de programas de Pc y ver sus posibilidades,pienso que es un programa importante dentro del software del QL,pero siempre que lo consideremos un juguete que nos va a abrir la ventana que asoma al mundo de los PCs.

Comparandolo con el Transformer de Amiga,mas o menos son igual de veloces,pero "The Solution" tiene la enorme ventaja que soporta gráficos aunque su manejo es tan lento que además de desesperante,deja de ser práctico.Pero ya es una enorme ventaja,por ejemplo,poder cargar el Dbase III Plus y manejar el programa en que están hechos la mayoria de los programas de gestion industrial y comercial a pequeña escala.O el procesador de textos DW3,o el OPEN ACCESS II,o tantos otros programas míticos del mundo del Pc.Ya no solo es de oidas sino que podemos cargarlos y lentamente manejarlos,hacernos idea del funcionamiento y de su calidad.

Pero no nos engañemos ni saquemos los pies del cesto.No estamos ante ninguna solucion.Si alguien cree que gastandose el dinero del programilla va a tener un PC en su casa está muy equivocado.Es algo asi como el que se comprara un ZX81 para cubrir las necesidades informaticas de su empresa.Repito,estamos ante un juguete muy bonito que nos puede proporcionar horas de entretenimiento y solaz,asomarnos al mundo de los PCs y no tener que hablar de oidas.

Yo tengo en la empresa que trabajo 2 Pcs,un Olivetti 240 con un h.d. de 20 Megas y floppy de 5 1/4 y otro Pc Philips 9002 con h.d. de 40 M. y flopys de 3.5 y 5 1/4.Como el software de aplicación en el area que dirijo está desarrollado integramente por nosotros a base de Dbase III Plus,Multiplan,Word Perfect etc y teniendo que ir a la oficina muchas tardes por razones de trabajo,ya que alli están los Pcs. decidí para Navidad,y por razones obvias,comprarme un Pc.Y asi lo hice.Tengo un INVES PC-X30 Turbo con floppys de 3.5 y 5 1/4 edemás de un H.D. 30 Megas,monitor de color CGA,ratón,impresora,mesa etc..¡la locura!.No quiero ni pensar lo que me costó pero ahí está.Fué una inversión.Por supuesto lo pongo todo ello a vuestra entera disposición y sin reservas para lo que necesiteis.

Os cuento ésto porque pienso que tengo los medios necesarios para poder hacer un estudio bastante eficaz de las cualidades de "The Solution" sin apasionamientos.

En el mundo de los Pcs,al haber tantisimas marcas de clonos,perifericos y configuraciones,las cualidades de cualquier Pc se miden ponderando una serie de factores comparandolos con el original IBM PC-XT que sirve como patrón.Hay por ello unos programas que además de mostrar ls configuración interna,memoria,sistema operativo,etc.etc. del Pc a estudiar,le hacen una serie de pruebas tanto de calculos,transferencia de datos,etc.etc. y al final dan un número que expresa su velocidad ponderada sobre el XT.

Yo he pasado "The solution" por tres de ellos,posiblemente los mas serios y extendidos.

Estos han sido:PCTOOLS PLUS R4.30(algo parecido a nuestro Toolkit II)

SYSTEM INFORMATION DE U.NORTON-87
INDICE DE PRESTACIONES de Microway R. 1986

Tanto en el Pctools como en el Norton no os reflejo la información sobre el Inves porque no viene a cuento salvo que en el indice de prestaciones da para él el indice 100% en modo XT y 200% en modo Turbo(Conmutador exterior que pasa la velocidad del reloj de 4.7 a 10 Mg).

A continuacion os pongo la información que dió Pctools sobre el "PC" del que tenia que emitir diagnostico.Hay cosas curiosas.

Se me olvidaba decir que The Solution corria con un DOS 3.30 de Memorex lo cual no creo tenga la mas minima importancia.

```
               Sistema operativo : DOS 3.30
                  Numero drives : 5
             Letras para drives : A a E
                  Puertas serie : 1 ¿.?
               Puertas Paralelo : 1 ¿.?
                    Tipo de CPU : NEC V20   ¿Y ésto de dónde sale?
      Velocidad relativa (XT=100) : 0 (No llega a medirla)
          Coprocesador matematico : No
         Memoria usada por DOS en
         programas residentes.... : 51.920 Kbytes
           Total memoria para DOS : 481 Kbytes
         Pctools ha encontrado un
         total de memoria RAM.... : 592 Kbytes
```

Después cargué el SI.EXE (System Information) de Norton Utilities-87 y el resultado de la exploración es significativamente parecido:

```
              Sistema operativo : DOS 3.30
                    Tipo de CPU : NEC V20 (¡Coincide!)
                  Puertas serie : 1
               Puertas paralelo : 1
                Adaptador Video : Color/Graficoos
              Actual modo video : Texto,80 X 25 Color
                         Drives : 2 , A y B
               Memoria para DOS : 481 Kbytes
          Memoria usada  por DOS
         y programas residentes : 50 Kbytes
                  Memoria libre : 431 Kbytes
                  Memoria total : 590 Kbytes
               Memoria pantalla : 96 kbytes
              Memoria uso extra : 75 Kbytes

    INDICE RELATIVO A PC/XT (XT=1) : 0.1
```

Es curioso que en el tipo de CPU coincidan los dos programas.Yo no conozco este tipo de CPU llamado V20 de NEC pero pienso seria uno de los primeros,muy lento de por si (¿quizás una CPU de gas-oil?).En el resto las apreciaciones son casi identicas.

Después cargué el programa INDICE DE PRESTACIONES de Microway R.y a continuación os presento un estudio comparativo con el INVES tanto en modo XT como TURBO.

Indice de Prestaciones	XT	TURBO	SOLUTION
Operaciones Aritmeticas	0.97	1.97	0.04
Operaciones Trascendentes...............	0.98	1.98	0.04
Bus de Datos:4 Kbytes de instrucciones...	0.97	1.96	0.02
Bus de Datos:4 Kbytes transf.registros...	0.97	1.96	0.03
Bus de Datos:16 Kbytes de instrucciones..	0.97	2.00	0.02
Reloj:Transf.de bloques de 4 kbytes......	1.00	2.01	0.07
Reloj:Transf.de bloques de 16 Kbytes.....	1.00	2.01	0.07
INDICE TOTAL PONDERADO (XT = 1)......	0.98	1.98	0.03

En éste último no incluye en la ponderación la velocidad de acceso a drives ni perifericos en general,solo velocidad pura de la CPU y circuitos asociados.

Como vereis la calificación en el mejor de los casos es de un 10% referido al mas lento de los PCs(el XT) y de un 0.03 en velocidad pura.Esto os puede dar

una idea de las prestaciones de "The Solution".

Estoy a punto de recibir el programa PROCOMM,posiblemente el mejor programa de comunicaciones escrito nunca para un PC,y voy a utilizar "The Solution" para acceder a Bases de Datos.En el próximo boletin os contaré la experiencia.La idea es la de utilizar el Inves en modo HOST y el QL como terminal.Podria ser el inicio de una base de datos especializda en QL con acceso via moden half duplex 1200 baudios.

¡Hasta la próxima.....!

Antonio Rodriguez Hdz, 22-5-89

PROGRAMA : TexTIDY
DISTRIBUIDOR : PDQL
AUTOR : D. J. WALKER

Se acuerdan del comentario de IBMCOPY, pues da la casualidad que este es el mismo autor del DISCOVER (versión actualizada del IBMCOPY), el cual ha escrito también el TEXTIDY como complemento del Discover/IBMCOPY.

TexTidy ha sido diseñado para resolver problemas que se pueden encontrar moviendo ficheros entre diferentes programas de una misma máquina o entre similares programas en diferentes diferentes máquinas.

Las facilidades de TEXTIDY son:

a) Ficheros de texto que contienen información sobre control pueden ser limpiados de esa información a simples ficheros ASCII. Estos ficheros pueden terminar en LF (Standard QL) o CR/LF (Standard CP/M y PCDOS/MSDOS). Esta última opción es usada también en los ficheros export de los programas de Psion.

b) Documentos QL QUILL y DOS QUILL pueden ser convertidos a ficheros ASCII simples.

c) Documentos Wordstar pueden ser convertirdos a simples ficheros ASCII . Tales ficheros pueden ser leidos por programas como THE EDITOR o importandolos dentro de QL QUILL.

d) Documentos QL QUILL y DOS QUILL pueden ser convertidos a ficheros Wordstar. Esta conversión contiene información como negrita, bajo.

En resumidas cuentas, TexTIDY es un traductor de textos que nos adapta cualquier documento QL a PC o viceversa.

TEXTIDY es incompatible o tiene problemas con Speedscreen, Lightning y TOOLKIT III (con confundir con TOOLKIT II).

TEXTIDY corre en multitarea. Usa menus y es muy fácil de usar (no hace falta leerse las instrucciones).

S. Merino, Fuengirola, 7/6/1989.

LENGUAJE FORTH

PRACTICANDO CON EL SUPERFORTH (XIII)

SDUMP CON LA TRUMPCARD

En un articulo anterior escribí de pasada que era posible usar el SDUMP desde el Superforth sin tener que echar mano al ensamblador. Pues he ahí una pequeña golosina :

```
5 STRING *SDUMP
2VARIABLE #SDUMP
*SDUMP READ" SDUMP"
CREATE SDP_SET 1 C, 1 C, 2 C, 1 C,
CREATE BLOQUE_VENTANA 256 , 240 , 0 , 0 ,

: WINDOW  BLOQUE_VENTANA 13 #SDUMP 2@ 0 TRAP3 2DROP DROP ;

: SDUMP 2 *SDUMP OPEN_DEVICE  #SDUMP 2!
    SDP_SET 7 #SDUMP 2@ 4 TRAP3 2DROP DROP
    WINDOW
```

```
    #SDUMP 2@ CLOSE ;

END_FILE
```

SDUMP se activa después de una definición WINDOW. El valor destino por defecto es SER.

Se podria haber definido una palabra introducir nuevos valores para las ventanas.

```
    CREATE BLOQUE_VENTANA 8 ALLOT

    : #BV ( ANCHO, ALTO, X, Y )
           BLOQUE_VENTANA 6 + 4 0 DO DUP >R ! R> 2- LOOP DROP ;
```

También se puede obtener la información necesaria para hacer un SDUMP de una determinada ventana usando SD.PXENQ.

```
    : SD_PXEND  ( bloque_canal, #canal, -- )
                 10 ROT ROT 0 TRAP3 -6 ?ERROR 2DROP ;

    ( #canal es el número de 32 bit ID del canal )
```

Para hacer un SDUMP a un dispositivo que no sea SER, añadir a SDP_SET suficientes bytes libres para guardar el código 9, byte longitud cadena y cadena ASCII del dispositivo. Luego solamente hay que hacer una operación IO.SSTRG (TRAP3 D0=7) con los Bytes a enviar.

Traducir esto a Assembler está chupado, lo dejo como entretenimiento.

Ya no sé si seguir con el rollo de los capitulos o utilizar un titulo que tenga algo que ver con el contenido e, incluso creo que me he desviado un poco del tema que estaba escribiendo últimamente (¡Lo que son las cosas!).

Tengo una pequeña idea de donde venimos, pero no tengo ni idea de a donde vamos (y se supone que soy yo quien debe saberlo), pero lo importante es que aún estamos vivos para contar algo.

S. Merino, 10/5/89.

CONVERTIR LA UTILIDAD TRANS-BASE EN STAND-ALONE

Cuando me visitó José Carlos de Prada e intentó mostrarme con un aire de lo nunca visto su utilidad Trans-Base, comprendí que al programa le faltaban algunos pequeños detalles. Aunque cargarlo fue un SHOW, el cual debo censurar, debo admitir que quedé sorprendido por la sencillez del programa, pues por un momento crei que José Carlos se habia entretenido en mejorar mi anterior calculadora, lo cual era bastante lógico, pues aún estoy esperando que alguien la perfeccione a su gusto.

El programa Trans-Base está bien diseñado, pero tiene un pequeño defectillo. La versión suminitrada por José Carlos solamente puede ser usada por miembros de CUQ que posean una copia del SUPERFORTH, por lo que su interés puede quedar mermado.

Para que el programita sea útil deberia haberse suministrado una versión tipo programa comercial usando la utilidad del BLK7, pero el programa original no sirve para correr como Stand-Alone. ¿Porqué no sirve? Muy simple, los canales están abiertos en modo intérprete en vez de compilado, y el cursor no está activo (indispensable en programas multitarea sin Qram, Taskmaster, etc..).

```
( ****** Trans-Base  v 1.0    Abril 1989   José Carlos de Prada ******* )
        ( Adaptado a versión Stand-Alone por MERINO-SOFT 13/5/89 )

          DECIMAL
          2VARIABLE #BASE10    ( variables para los id de los canales )
          2VARIABLE #BASE2
          2VARIABLE #BASE4
          2VARIABLE #BASE8
          2VARIABLE #BASE16
          2VARIABLE #CONSOLA

          31 STRING NUMERO          ( variable de cadena para entrada )
          2VARIABLE NUM            ( variable para entrada convertida )
          2.147483647  2CONSTANT  MAXIMO               ( valor máximo )
          18 STRING #T0
          18 STRING #T1
```

```
          18 STRING #T2
          18 STRING #T3
          18 STRING #T4
          18 STRING #T5
          #T0 READ" CON_240X72A258X20"
          #T1 READ" CON_192X12A300X22"
          #T2 READ" CON_192X12A300X36"
          #T3 READ" CON_192X12A300X50"
          #T4 READ" CON_192X12A300X64"
          #T5 READ" CON_192X12A300X78"

: ABRIR_CANALES
               4 MODE
               0 #T0 OPEN_DEVICE #CONSOLA 2!
               0 #T1 OPEN_DEVICE #BASE10 2!      ( abrir canales y  )
               0 #T2 OPEN_DEVICE #BASE2  2!      ( guardar su id    )
               0 #T3 OPEN_DEVICE #BASE4  2!      ( en las variables )
               0 #T4 OPEN_DEVICE #BASE8  2!
               0 #T5 OPEN_DEVICE #BASE16 2!  ;

: #IO     ( d -- ) ( pone la entrada/salida en el canal guardado en d )
          2@ 2DUP  #IN 2!  #OUT 2! ;
: LETREROS ( -- )                               ( rotula los letreros )
          #CONSOLA #IO
          7 PAPER 7 STRIP 2 INK 0 0 CSIZE 255 1 BORDER  CLS
          4  2 CURSOR ."  DEC"
          4 INK
          4 16 CURSOR ."  BIN"
          4 30 CURSOR ."  B-4"
          4 44 CURSOR ."  OCT"
          4 58 CURSOR ."  HEX" ;
: VENTANA  ( fondo papel col-borde ancho-borde tinta -- )
          INK BORDER PAPER STRIP CLS ;    ( parámetros de una ventana )
: VENTANAS ( -- )                   ( pone parámetros de las ventanas )
          LETREROS
          #BASE10    #IO 2 2   4 1 7 VENTANA
          #BASE2     #IO 4 4   2 1 0 VENTANA
          #BASE4     #IO 4 4   2 1 0 VENTANA
          #BASE8     #IO 4 4   2 1 0 VENTANA
          #BASE16    #IO 4 4   2 1 0 VENTANA  ;
: SALIDA  ( n ad -- )    ( escribe la salida en el canal ad en base n )
          #IO CLS NUM 2@ ROT BASE ! 31 D.R ;
: BN      ( n -- )           ( convierte una entrada al resto de bases)
          DECIMAL  BASE !
          CLS
          CURSOR_ON NUMERO INPUT CURSOR_OFF    ( lee la entrada )
          0. NUMERO CONVERT NUMERO COUNT + >=   ( conversión a número )
          IF
               2DUP MAXIMO 2SWAP D<           ( error de rebosamiento )
               IF
                    2DROP CLS ."           REBOSAMIENTO"
               ELSE
                    NUM 2!                          ( salida correcta )
                    10 #BASE10 SALIDA
                    16 #BASE16 SALIDA
                    8  #BASE8  SALIDA
                    4  #BASE4  SALIDA
                    2  #BASE2  SALIDA
                THEN
          ELSE                                ( error de tipo de dato )
                CLS ."         ERROR DE ENTRADA"
          THEN
          DECIMAL ;
: B10     ( -- )                                  ( entrada en base 10 )
          #BASE10 #IO 10 BN ;
: B2      ( -- )                                  ( entrada en base  2 )
          #BASE2  #IO  2 BN ;
: B4      ( -- )                                  ( entrada en base  4 )
          #BASE4  #IO  4 BN ;
: B8      ( -- )                                  ( entrada en base  8 )
```

```
          #BASE8  #IO  8 BN ;
: B16     ( -- )                                  ( entrada en base 16 )
          #BASE16 #IO 16 BN ;
: ELIGE                              ( selección de la base de entrada )
          2 2 2DUP #IN 2! #OUT 2! 0 0 AT CURSOR_ON KEY CURSOR_OFF
             CASE
                    68 OF B10 0 ENDOF
                   100 OF B10 0 ENDOF
                    66 OF B2  0 ENDOF
                    98 OF B2  0 ENDOF
                    67 OF B4  0 ENDOF
                    99 OF B4  0 ENDOF
                    79 OF B8  0 ENDOF
                   111 OF B8  0 ENDOF
                    72 OF B16 0 ENDOF
                   104 OF B16 0 ENDOF
                    27 OF     1 ENDOF
                   248 OF VENTANAS ENDOF
                    DEFAULT 100 500 BEEP 0
          ENDCASE ;
: TRANSBASE   ( -- )                             ( programa principal )
          ABRIR_CANALES
          VENTANAS
          BEGIN
                    ELIGE
          UNTIL
          BYE ;
END_FILE
```

En el listado anterior podreis observar todos los cambios que he hecho para que el programa pueda ser salvado como un Stand-Alone.

El cursor que veis en la esquina izquierda de la pantalla corresponde a cuando tenemos que elegir una opción. Existe una nueva opción : F5 redibuja la pantalla.

Todos los programas se pueden mejorar, quizás un programa nunca está definivamente terminado.

Traducir el Trans-Base a Assembler es algo casi directo, y su mayor ventaja es que ocuparia menos de 1Kb (en Forth tenemos que agregar el diccionario y por eso ocupa unos 18K).

S. Merino, Fuengirola, 13/5/89.

Z88

INSTALANDO LA IMPRESORA EN UN Z88

Si hay algo interesante aún en mi vieja BMC es que el cable RS-232 es más barato que la opción Centronic, y que a pesar de sus años funciona todavia con mi viejo Spectrum y, como no, nuestro admirable QL. Pero se necesitan 3 cables totalmente diferentes, uno para cada ordenador (¡ésto suena a cañondeo!).

El Z88 Español se suministra con manual en Inglés y si somos exigentes, nos regalan una versión del manual en Español, la cual no se sabe ni de donde porras han traducido (el manual en Inglés es más completo).

El Z88 Español posee un teclado Español (¡faltaria más!) que a diferencia del Inglés, posee más caracteres y tiene acceso fácil a los caracteres ingleses sustituidos (están escritos en rojo). La única pega es que los programas vienen en Inglés, pero yo creo que es incluso mejor que sea asi, pues se evitan lios a la hora de consultar la biblia del Z88 (Z88 Developer notes (Cambridge Computer), todo lo que debemos saber sobre el sistema operativo OZ (ya os comentaré el libro en otra ocasión))

Lo primero que ocurre es que uno piensa que la instalación de la impresora que viene de fábrica, al ser tipo Epson no hay que hacer nada. ¡Qué equivocados estamos! La instalación está programada para que cada vez que se imprima una

página, haga un FF (comenzar en el principio de la nueva página). Esto con hojas sueltas es un rollo, hay que eliminarlo. Después resulta que en el manual dice que solamente se pueden hacer 9 traducciones, pero ¿en qué mundo vivimos?. UFF... existe una nueva opción, ISO Translations, que nos permite traducir todo el juego de caracteres por encima de 128, pero en el manual no vienen en la tabla los nuevos caracteres. No hay problema, tenemos el BBC BASIC para averiguarlo (en Cambridge deben creer que la máquina es solamente para Hackers).

DECIMAL	ASCII	TRADUCCION PARA BMC
161	¡	91
163		27,82,13,35,27,82,17
191	¿	93
199	Ç	
209	Ñ	92
224	à	
225	á	97,8,39
226	â	
231	ç	
232	è	
233	é	101,8,39
234	ê	
236	ì	
237	í	105,8,39
238	î	
239	ï	
241	ñ	124
242	ò	
243	ó	111,8,39
244	ô	
246	Ö	
249	ù	
250	ú	117,8,39
251	û	
252	ü	117,8,12

Mi impresora BMC está instalada por hardware con el juego Español. Solamente he traducido los caracteres que más uso.

Es muy posible que este articulo no se pueda imprimir en la impresora correctamente sin la ayuda de un programa traductor tipo OVERDRIVE (hay muchas versiones donadas por programadores a QUANTA u otros grupos).

página 1

S. Merino, Fuengirola, 28/5/89.

13 Conversiones de Formato

El Toolkit II aporta un número de facilidades para entradas y salidas en un formato fijado. Entre otras, se encuentran las conversiones de números binarios y hexagesimales así como el formato decimal fijo. La mayoría de estas facilidades son en forma de función, pero se incluye también un nuevo comando.

13.1 PRINT_USING

PRINT_USING es una versión de **PRINT** para la escritura en formato fio.
PRINT_USING ***#canal,formato,lista de items a escribir***
El "formato" es una cadena o expresión de cadena que contiene una imagen de la salida deseada. En dicha cadena, los caracteres **+ - # * , . ! " ' $** y @ tienen un significado especial. El procedimiento escribe los caracteres de la cadena "formato" hasta que encuentra uno especial.
Si se encuentra el caracter @, el siguiente caracter será escrito aunque se trate de un caracter especial.
Si se encuentra un caracter " o ' (comillas), se escribirán todos los caracteres que siguan hasta encontrar otras comillas.
Si se encuentra el caracter **\b0 , se escribirá un caracter de "nueva linea".**
El resto de caracteres aparecen en "campos". Para cada campo, se coge un ítem de la lista, se le formatea de acuerdo a la forma del campo y se le escribe.
El campo determina no sólo la forma del ítem, sino también su ancho (igual al ancho del campo). Las anchuras de los campos en los siguientes ejemplos son arbritarias:

c**ampo** **formato**
si el ítem es una cadena, escribirla justificándola a la izquierda o truncándola;
de lo contrario, escribir el número entero justificándolo a la derecha
***** Escribir el número entero justificándolo a la derecha y rellenando el campo con asteriscos (por ejemplo, ***12)
#####.## Escribir el número en formato decimal de coma fija (por ejemplo, 12.67)
****.** Escribir el número en formato decimal de coma fija, rellenando el campo con asteriscos (por ejemplo, **12.67)
##,###.## Escribir el número en formato decimal de coma fija, con los miles separados con comas (por ejemplo 1,234.56)
,*.** Idem, rellenándolo con asteriscos (por ejemplo *1,234.56)

-#.####!!!! Escribir el número en formato exponencial (por ejemplo, 2.9979E+08) con signo opcional
+#.####!!!! Idem, colocando siempre el signo
Un campo exponencial debe comenzar con un signo, un #, y un punto decimal (coma o punto). Debe acabar con cuatro signos de exclamación, **!!!!**.
Un campo decimal puede llevar como prefijo o como terminación un **+** o **-**, o puede ser incluido entre paréntesis. Si un campo está entre paréntesis, los valores negativos se escribirán entre paréntesis. Si se pone un **-**, el signo sólo se escribirá cuando sea negativo; si se pone un **+**, siempre se escribirá el signo. Si el **+** o **-** está al final del campo, el signo será colocado siguiendo al valor.
Los números pueden escribirse con un punto o con una coma para señalar la parte decimal. Si el campo sólo incluye un punto o una coma, dicho caracter será empleado como punto decimal. Si hay más de un punto o coma, el último de dichos caracteres será utilizado como punto (o coma) decimal y los otros como separadores de los millares. ¡ Larga vida a la unidad europea !.
Si el punto decimal está al final del campo, no será impreso. De esta manera, los valores monetarios pueden ser escritos con los millares separados pero sin punto decimal (por ejemplo 1,234).
El caracter **$** permite insertar en los campos , por ejemplo, los simbolos de las monedas de forma que sean colocados justamente antes del valor. Para ello debe colocarse el símbolo de la moneda entre el **$** y el primer # del campo (por ejemplo, **$**DM#.###,## o **$**$##.##).
Por ejemplo:

```
fmt$="Debe *******.**:($SKr##.###,##):##,###.##+\b0 "
PRINT_USING fmt$, 123.45, 123.45, 123.45
PRINT_USING fmt$, -12345.67, -12345.67, -12345.67
PRINT_USING "-#.###!!!!\b0 ", 1234567
```

escribirá

```
Debe ****123.45:    SKr123,45 :   123.45+
Debe *-12345.67:(SKr12.345,67):12,345.67-
 1.235E+06
```

13.2 Conversiones a Formato Decimal
Las siguientes rutinas convierten un valor en un número decimal en forma de cadena. El número de decimales es fijo y no se usa la forma exponencial.
FDEC$ (*valor,longitud,decimales*) formato decimal fijo
IDEC$ (*valor,longitud,decimales*)
CDEC$ (*valor,longitud,decimales*)
El número de caracteres de la cadena devuelta viene dado por el parámetro ***longitud***, mientras que el parámetro ***decimales*** señala el de decimales.
Las tres rutinas son similares. **FDEC$** convierte el valor tal cual, mientras que **IDEC$** asume que el valor dado es una representación

integral en unidades del dígito menos significativo mostrado. **CDEC$** es la conversión para monedas, siendo muy similar a **IDEC$** excepto en que se colocan comas cada tres dígitos.
FDEC$ (1234.56,9,2) devuelve " 1234.56"
IDEC$ (123456,9,2) devuelve " 1234.56"
CDEC$ (123456,9,2) devuelve " 1,234.56"
Si la longitud dada no es suficiente para contener el valor, la cadena se rellena con '*****'. El valor debe estar entre -2^31 y 2^31 (-2,000,000,000 a +2,000,000,000) para **IDEC$** y **CDEC$**, mientras que para **FDEC$** es el valor multiplicado por 10^***decimales*** el que debe estar en dicho rango.

13.3 Conversion a Formato Exponencial

La siguiente función permite convertir un valor en una cadena que representa al valor en forma exponencial.
FEXP$(*valor,longitud,decimales*)   formato exponencial fijo
La forma tiene un signo opcional y un dígito antes del punto decimal, y ***decimales*** dígitos después del punto decimal. El exponente se compone de una "**E**" seguida de un signo y de dos dígitos. La ***longitud*** debe ser al menos 7 unidades más grande que ***decimales***. Por ejemplo:
FEXP$(1234.56,12,4) devuelve " 1.23456E+03"

13.4 Números Binarios y Hexagesimales

HEX$(*valor,número de bits*)   convierte a hexagesimal
BIN$(*valor,número de bits*)   convierte a binario
Estas funciones devuelven una cadena de suficiente longitud como para representar el número de bits especificado de la parte menos significativa del valor. En el caso de **HEX$** el número de bits es redondeado al siguiente múltiplo de cuatro.
HEX(*cadena hexagesimal*) hexagesimal a valor decimal
BIN(*cadena binaria*) binario a valor decimal
Estas funciones convierten la cadena dada en un valor. Para **BIN**, cualquier caracter de la cadena cuyo valor ASCII sea par, será tratado como **0**, mientras que los caracteres con valor ASCII impar lo serán como **1**. Por ejemplo, **BIN('.#.#')** devuelve el valor **5**. Para **HEX**, los dígitos "**0**" a "**9**", "**A**" a "**F**" y "**a**" a "**f**" tienen su significado convencional; los demás caracteres generarán un error.

---

14 Control de la Pantalla

Las facilidades que se dan para el control de la pantalla son de tres tipos: control del cursor, control de la fuente de caracteres y reinicialización de las ventanas.

14.1 Control del Cursor

La función **INKEY$** fué diseñada para leer las pulsaciones sobre el teclado sin activar el cursor. Se dan dos procedimientos para conectar y desconectar el cursor. Un cursor conectado suele aparecer como un rectángulo sólido (estado inactivo). El cursor comenzará a parpadear (estado activo) cuando la cola del teclado se encuentre conectada a la ventana que contiene el cursor (por ejemplo, mediante un **INKEY$**).
CURSEN ***#canal*** conecta el cursor
CURDIS ***#canal*** desconecta el cursor
Observa que mientras **CURSEN** y **CURDIS** utilizan, como la mayoría de comandos de entrada/salida, el canal **#1** como canal de omisión, **INKEY$** utiliza el #**0**.
Por ejemplo:
CURSEN:in$=**INKEY$**(#1,250):**CURSEN**
conectará el cursor en la ventana #1, y esperará 5 segundos a que sea pulsado un caracter desde el teclado. Si no se pulsa nada en el teclado, in$ tomará un valor de cadena nulo ("").

14.2 Control de la Fuente de Caracteres
El controlador de la pantalla del QL tienen dos fuentes de caracteres en su interior. La primera fuente provee los patrones para los caracteres 32 (espacio) a 127 (copyright), mientras que la segunda provee los patrones para los caracteres 127 (no definido) a 191 (flecha hacia abajo). Para cada caracter, el controlador de pantalla utiliza el patrón apropiado de la primera fuente; si no lo encuentra en la primera fuente, lo busca en la segunda, y en caso de que tampoco lo encuentre en la segunda, usará el primer patrón definido en esta segunda fuente.
Las fuentes sustitutas no tienen porqué tener el mismo rango de caracteres que las fuentes internas. Puede, por ejemplo, definir una fuente que tenga todos los valores desde el 128 al 255.
El formato de una fuente del QL es:

1 byte que indica el caracter más bajo de la fuente
1 byte que indica el número de caracteres válidos - 1
9 bytes con el patrón del caracter más bajo
9 bytes con el patrón del siguiente caracter
etc.

Los caracteres estándard del QL se componen de 9 líneas de puntos de pantalla, cada una de 5 puntos de ancha. La primera línea de puntos de pantalla que compone el patrón del caracter se almacena en el primer byte y así hasta la novena línea que lo hace en el noveno byte. Para cada línea, el bit 6 del byte correspondiente indica el estado del punto situado más a la izquierda de la línea y el 2 el del más a la derecha. Un bit igual a uno, indica que debe colorearse con el color de la tinta (**INK**), mientras que cero que debe colorearse con el color del papel (**PAPER**).
El caracter '**g**' se almacena como: **%00000000**

%00000000
%00111000
%01000100
%01000100
%01000100
%00111100
%00000100
%00111000

El comando **CHAR_USE** se utiliza para cambiar una o las dos fuentes de caracteres.

CHAR_USE *#canal,dirección1,dirección2* "dirección1" y "dirección2" son las direcciones de memoria donde se encuentran las fuentes sustitutas

CHAR_USE *#canal,*0*,dirección2* como primera fuente será utilizada la interna del QL

CHAR_USE 0,0 reinicializa ambas fuentes para la ventana #1

El controlador de la pantalla del QL asume que todos los caracteres son de 5 puntos de pantalla ("pixels") de ancho por 9 de alto. El resto de tamaños se obtienen duplicando los puntos o colocando puntos en blanco entre los caracteres. Con el Toolkit II es posible escoger cualquier espaciado horizontal y vertical. Si el incremento se selecciona a un valor más pequeño que el del tamaño actual de los caracteres (seleccionado mediante **CSIZE**), deberá tenerse extremo cuidado ya que es posible que el controlador de la pantalla escriba caracteres (en la parte derecha o en la inferior) parcialmente fuera de la ventana: la distancia de las ventanas a los límites derecho o inferior de la pantalla no deberá ser inferior a la diferencia entre el tamaño de los caracteres (seleccionado con **CSIZE**) y el espaciado de los mismos (seleccionado con **CHAR_INC**).

CHAR_INC *#canal,incremento x, incremento y* selecciona los espaciados x (ancho) e y (alto)

El canal de omisión es el **#1.**

Los espaciados de caracteres especificados mediante el comando anterior son anulados cuando se emplea un comando **CSIZE.**

Por ejemplo, si hay una fuente de caracteres 3x6 en un fichero llamado "f3x6" (de longitud 875 bytes), se puede seleccionar una pantalla de 127 columnas por 36 líneas mediante los siguientes comandos:

```
MODE 4
WINDOW 512-2,256-3,0,0   :REMark límites de la ventana
CSIZE 0,0 :REMark espaciado 6x10
CHAR_INC 4,7   :REMark espaciado 4x7
fount=ALCHP(875)    :REMark reserva espacio para fuente
LBYTES f3x6,fount   :REMark carga la fuente
CHAR_USE fount,0    :REMark selecciona la nueva fuente como fuente primera
```

14.3 Reinicializando las Ventanas
Hay dos procedimientos para reinicializar las ventanas al estado que tienen al encender el aparato:
WMON *modo* reinicializa a "Monitor"
WTV *modo* reinicializa a "Televisión"
El modo debe ser 0, 4 o 512 para el modo de alta resolución, u 8 o 256 para el de baja resolución. Sólo se reinicializan los tamaños, posiciones y bordes de las ventanas pero no el color del papel o de la tinta.

--

15 Utilización de la Memoria

Como el QDOS es un sistema operativo multitarea, y puede haber varias tareas ejecutándose a la vez en un QL, la cantidad de memoria libre es impredecible. Ninguna tarea debe asumir que la cantidad de memoria libre es fija. Puede usarse la función **FREE_MEM** para averiguar la cantidad de memoria libre (definida como el espacio disponible para el sistema de bloques esclavos de ficheros menos el espacio requerido para dos -en el QL Toolkit sólo uno- de dichos bloques).
Se puede reservar temporalmente espacio en la "pila común" mediante la función **ALCHP** la cual devuelve la dirección de la base del espacio reservado. Se pueden anular reservaciones individuales mediante el comando **RECHP**, o se pueden anular todas mediante los comandos **CLCHP** (limpia la pila común), **CLEAR** o **NEW.**
FREE_MEM devuelve la cantidad de memoria libre
ALCHP (*número de bytes*) reserva memoria en la pila común y devuelve la dirección de la base del espacio
RECHP *dirección de la base* libera espacio de la pila común
CLCHP libera todas las reservas de la pila común
Si se hacen grandes reservas en la pila común y, a continuación, se accede por primera vez a un disco, se puede causar un terrible mal a la pila llamado "fragmentación en gran escala": los bloques de definición del dispositivo se dispersan enormemente en la pila dejando grandes agujeros que dejan de estar disponibles excepto para entradas de la pila (es decir, en ellos no se pueden cargar programas). Un solución simple pero peligrosa consiste en borrar los bloque de definición de los dispositivos:
DEL_DEFB borra fichero de los bloques de definición de la pila común
Aunque en dicho procedimiento toma precauciones para minimizar el daño, debe tenerse cuidado de evitar el uso de este comando cuando algún dispositivo de tipo directorio esté activo.

--

16 Parámetros de los Procedimientos

Los parámetros de los procedimientos del SuperBasic son utilizados mediante sustitución: al llamar al procedimiento (o función), los falsos parámetros de la definición del procedimientos se convierten en los parámetros reales en la llamada del procedimiento. El tipo y uso de los parámetros del procedimiento puede averiguarse mediante dos funciones:

PARTYP (*nombre*) devuelve el tipo de parámetro
PARUSE (*nombre*) devuelve el uso del parámetro

El tipo es 0 nulo El uso es 0 no escogido
1 cadena 1 variable 2 coma flotante 2 matriz 3 entero

Uno de los trucos usados por algunos procedimentos en código máquina consiste en usar el "nombre" de un parámetro real en lugar de su valor (por ejemplo, "LOAD fred" cargará un fichero llamado "fred"). Dado el nombre de un falso parámetro de un procedimiento, sería posible encontrar el nombre de un parámetro real de una llamada a un procedimiento del SuperBasic, pero sería bastante lento. Es mucho más fácil encontrar el nombre de un parámetro real, si se conoce su posición en la lista de parámetros.

PARNAM$(*número de parámetro*) devuelve el nombre del paramétro

Por ejemplo, el fragmento de programa

```
pnombre fred,joe,"mary"
DEFine PROCedure pnombre(n1,n2,n3)
PRINT PARNAM$(1),PARNAM$(2),PARNAM$(3)
END DEFine pnombre
```

imprimirá "fred   joe   " (la expresión no tiene nombre).

Otro truco consiste en usar el valor del argumento verdadero si es una cadena, o el nombre en caso contrario. Ello es posible en procedimientos del SuperBasic usando la función **PARSTR$**:

PARSTR$ (*nombre, número de parámetro*) si el parámetro "nombre" es una cadena, devuelve el valor, sino encuentra su nombre

Por ejemplo, el fragmento de programa

```
pcadena fred,joe,"mary"
...
DEFine PROCedure pcadena(n1,n2,n3)
PRINT PARSTR$(n1,1),PARSTR$(n2,2),PARSTR$(n3,3)
END DEFine pcadena
```

imprimirá "fred   joe   mary".

----

17 Manejo de Errores

Las ROMs **JS** y **MG** del QL contienen el código inacabado para la captura

de errores en el SuperBasic: El Toolkit II corrige algunos de los problemas existentes.
El manejo de errores se invoca mediante una cláusula **WHEN ERROR.** Contrariamente a las definiciones de procedimientos y funciones, estas cláusulas son estáticas. El manejo de errores es activado cuando se encuentra una cláusula **WHEN ERROR**, pero sólo es accionado cuando (**WHEN**) se encuentra un **ERROR.** Ello significa que un programa puede tener más de una cláusula **WHEN ERROR.** Cada vez que una de ellas es ejecutada, el proceso de errores definido en la cláusula reemplaza al anteriormente definido.
La cláusula se abre con una sentencia **WHEN ERROR** y se cierra con una **END WHEN.** Dentro de la cláusula puede haber cualquier tipo de sentencias (¡aunque es conveniente evitar la llamada a funciones o procedimientos del SuperBasic!). De una cláusula **WHEN ERROR** se sale mediante los comandos **STOP**, **CONTINUE**, **RETRY**, **RUN**, **LOAD** o **LRUN** (siempre que se esté usando el Toolkit II). Además, las versiones de **RUN**, **NEW**, **CLEAR**, **LOAD**, **LRUN**, **MERGE** y **MRUN** en el Toolkit II reinicializan el proceso de errores (una desafortunada omisión en las ROMs del QL).
Hay una serie de facilidades adicionales pensadas para su uso en las cláusulas **WHEN ERROR.**

17.1 Errores

Las siguientes funciones se corresponden con cada uno de los códigos de error del sistema:

ERR_NC	**ERR_NJ**	**ERR_OM**	**ERR_OR**	**ERR_BO**
ERR_NO	**ERR_NF**	**ERR_EX**	**ERR_IU**	**ERR_EF**
ERR_DF	**ERR_BN**	**ERR_TE**	**ERR_FF**	**ERR_BP**
ERR_FE	**ERR_XP**	**ERR_OV**	**ERR_NI**	**ERR_RO**
ERR_BL				

y devuelven el valor verdadero si el error que causó la invocación de la cláusula **WHEN ERROR** es del tipo dado. ***No*** usar **ERR_DF** sin el Toolkit II.
Los siguientes procedimientos dan información sobre el error producido:
ERLIN devuelve el número de línea en el que ocurrió el error
ERNUM devuelve el número de error
REPORT *#canal* informa del último error
REPORT informa del último error en el canal #0
REPORT *#canal,número de error* informa del error cuyo número es el especificado

17.2 RETRY y CONTINUE

Ya que **RETRY** y **CONTINUE** permiten salir de una cláusula de error sin reinicializar el **WHEN ERROR**, sería útil que pudieran ser también usados para salir a una parte distinta del programa. En el Toolkit II, **RETRY** y **CONTINUE** pueden tener un número de línea.
CONTINUE *número de línea* continuar desde la línea dada

RETRY *número de línea* "retry" desde la línea dada

```
100 WHEN ERRor
110 IF ERLIN=200: PRINT #0\'¡ojo!',:RETRY
120 REPORT
130 STOP
140 END WHEN
150 :
160 do_in x
170 STOP
180 DEF PROCedure do_in(j)
190 FOR i=1 TO 10
200 INPUT #0,'entrada';j
210 PRINT #0,'valor';j
220 END FOR i
230 END DEFine do_in
```

---

18 Mantenimiento del Tiempo

18.1 Reloj Digital Residente

CLOCK reloj de omisión en su propia ventana
CLOCK *#canal* reloj de omisión, 2 filas de 10 columnas
CLOCK *#canal,cadena* reloj definido por el usuario
CLOCK es un procedimiento que selecciona un reloj digital residente. Si no se especifica ninguna ventana, se selecciona una ventana de omisión en la esquina superior derecha de lo que es la ventana de omisión #0 en el modo monitor. Esta ventana es de 60x20 "pixels" (puntos de pantalla) y es sólo apropiada para el modo 4. Se puede poner el reloj en una ventana dada del SuperBasic. En este caso, el reloj será eliminado cuando se cierre la correspondiente ventana.
El usuario puede definir también un reloj a su conveniencia. El parámetro "cadena" indica los caracteres que se escribirán en la ventana del reloj. Se puede escribir cualquier caracter salvo **$** y **%**.
El procedimiento, cuando encuentra un **$** en la cadena, mira el caracter que le sigue y si es
$d o **$D** inserta en su lugar los tres caracteres del día de la semana,
$m o **$M** inserta en su lugar los tres caracteres del mes.
Cuando encuentra un **%**, mira el caracter que le sigue y si es
%y o **%Y** inserta los dos dígitos del año
%d o **%D** inserta los dos dígitos del día del mes
%h o **%H** inserta los dos dígitos de la hora
%m o **%M** inserta los dos dígitos del minuto
%s o **%S** inserta los dos dígitos del segundo
La cadena del reloj de omisión es '**$d %d $m %h/%m/%s '.** Un salto de

línea debe ser forzado rellenando la línea de espacios hasta alcanzar el margen derecho de la línea.
Ejemplo:

```
MODE 8
OPEN #6,scr_156x10a32x16
INK #6,0 : PAPER #6,6
CLOCK #6,'Hora QL %h:%m'
```

18.2 Reloj-Alarma

ALARM *hora* selecciona el reloj-alarma para que suene a una determinada hora
La hora debe ser especificada como dos números: horas (de 0 a 24) y minutos.
ALARM 14,30 la alarma sonará a las dos y media

---

19 Extras

EXTRAS *#canal* lista las facilidades extras añadidas al SuperBasic
EXTRAS lista los extras en el canal #1
Si el canal de salida es una ventana, la pantalla se congela (CTRL F5) cuando la ventana se llena. Con el Toolkit II instalado, hay cientos de extras.
TK2_EXT en el caso de comandos con el mismo nombre en el Toolkit II que en otros Toolkits, este comando refuerza la definición que de dichos comandos se da en el Toolkit II frente a las otras
Si, por alguna razón, alguna de las extensiones del Toolkit II ha sido redefinida, **TK2_EXT** (compárese con **FLP_EXT** para las extensiones de las unidades de disco, **EXT_EXT** para las extensiones de las unidades de expansión) reafirmará la definición dada en el Toolkit II.

---

20 Controlador de Consola

20.1 Extensiones al Teclado

Hay dos extensiones al manejo del teclado del QL. La primera aporta la facilidad de recuperación de la última línea, y la segunda asigna una cadena de caracteres a la pulsación de <ALT> y otro caracter.
<ALT><ENTER> pulsación que recupera la última línea entrada
Esta pulsación recupera la última línea entrada, siempre y cuando el amortiguador del teclado sea lo suficientemente largo como para guardarla.

El comando **ALTKEY** asigna una cadena a una pulsación de "ALT" y otro caracter (mantenga la tecla "ALT" pulsada mientras presiona la otra). La cadena puede contener caracteres de nueva línea. Si se da más de una cadena, habrá un caracter de nueva línea implícito entre las cadenas. Para añadir un caracter de nueva línea al final de la cadena ponga una cadena nula ('' o "") al final de la línea.

ALTKEY *caracter,cadenas* asigna una cadena a una pulsación de <ALT>

Por ejemplo, después del comando

ALTKEY 'r','RJOB "SPL"',''

cuando se pulse **ALT** r, el comando **RJOB** "SPL" será ejecutado.

ALTKEY 'r' cancela la cadena ALTKEY para "r", mientras que

ALTKEY cancela todas las cadenas ALTKEY

---

21 Controlador de Microdrive

21.1 Extensiones a los Microdrives

Hay tres extensiones al sistema de ficheros en Microdrive. Están disponibles como puntos de entrada del sistema operativo, pero podrían ser soportados como llamadas desde el SuperBASIC.

OPEN OVERWRITE TRAP #2, D0=1, D3=3

Esta variante de la llamada OPEN abre un fichero para lectura/escritura exista o no exista. El fichero sa trunca a longitud cero antes de su uso.

RENAME TRAP #3, D0=4A, A1 apunta al nuevo nombre

Esta llamada renombra un fichero. El nombre debe incluir el nombre de dispositivo (por ejemplo, mdv1_nuevo_nombre)

TRUNCATE TRAP #3, D0=48

Esta llamada trunca un fichero a la posición del byte actual

21.2 Mejoras de los Microdrives

La llamada **FS.FLUSH** al sistema de ficheros efectúa un completo nivelado incluyendo la información de cabecera. Se puede acceder a esta operación mediante el comando **FLUSH**.

---

22 Controlador de Red Local

En el Toolkit II se ha intentado mejorar hasta un nivel útil las facilidades más bien elementales de red local en el QL.

El rendimiento de la red local se encuentra limitado por las capacidades excepcionalmente bajas del soporte físico ("hardware") de

la red local (Si su QL es de una versión anterior al número de serie D14, es altamente posible que el soporte físico de la red local no funcione en absoluto, aunque la experiencia reciente ha demostrado que hay más QLes pre-D14 de los supuestos con una puerta de red local que funcione).

22.1 Mejoras de la red local

Cada QL conectado a una red local debe tener un único número de estación en el rango de 1 a 63. Este número de selecciona con el comando NET.

NET *número de estación*

El toolhit II provee de un nuevo protocolo de emisión con nuevas aportaciones para la concordancia. Una emisión es un mensaje enviado desde un QL a todos los QLes a la escucha de la red local. El protocolo de emisión del Toolkit II tiene una concordancia positiva NACK (no reconocida, "not acknowledged"), además de previsión para la detección de BREAK.

Los nombres de dispositivo para la red local utilizan el siguiente convenio:

NETO_*número de estación* salida a un número de estación
NETO_0 enviar emisión
NETI_*número de estación* entrada de un núm. de estación
NETI_*mi núm.de estación* entrada desde cualquier estac.
NETI_0 recibir una emisión
NETI_0_*tamaño amortiguador* recibir una emisión en un amoriguador del tamaño especificado

Cuando se abre un canal para recibir una emisión, se abre un amortiguador para permitir la recepción ininterrumpida de la transmisión completa. Si no se especifica tamaño de amortiguador, entonces se toma toda la memoria libre a excepción de 2 K. EL tamaño del amortiguador debe ser especificado en Kbytes. Por ejemplo:

NETI_0_10 recibir emisión en un amortiguador de 10 Kbytes

Cuando se cierra un canal de salida de red local, el controlador de red local (de la misma forma que el controlador de red local del QL) mantendrá, tratando de enviarlo, el último amortiguador durante 20 segundos en el caso que la estación de espera esté trabajando con sus microdrives. Sin embargo, con el Toolkit II, pasados los 5 primeros segundos, el controlador empezará a revisar la pulsación de un BREAK.

22.2 Servidor de ficheros

El servidor de ficheros que se provee con el Toolkit II, es un programa que permite el acceso por parte de otro QL de 10 recursos unidos a un QL. Esto quiere decir, por ejemplo, que los dispositivos de disco unidos a un QL pueden ser accedidos por varios QLes diferentes. Sólo es necesario ejecutar el servidor de ficheros en el QL con los 10 recursos compartidos. Esta versión del servidor de ficheros es más general que la primera versión en el sentido que los 10 recursos pueden ser no

sólo dispositivos de fichero (p. ej. discos), sino también dispositivos en serie puros (p. ej. "modems" o impresoras) o ventanas de la pantalla del QL.

FSERVE invoca el servidor de ficheros

Puede haber más de un QL en una red local con un servidor de ficheros ejecutándose, pero el número de estación tiene que ser lo más baja posible y nunca mayor de 8.

Es posible que queden abiertos ficheros a través de la red local. Puede ocurrir así si se apaga o reinicializa un QL remoto. Para corregir esto, espera a que todos los QLes remotos hayan acabado sus operaciones en este QL, retira el servidor de ficheros

RJOB SERVER

y recomienza con el comando

FSERVE

22.3 Accediendo al servidor de ficheros

Un QL remoto accede a los ficheros en red local mediante un nombre de dispositivo compuesto:

N*número de estación_dispositivo de entrada/salida (IO)*
es el nombre de un dispositivo remoto de entrada/salida (p. ej.N2_FLP1_ es el disco 1 de la estación 2 de la red local)

Por ejemplo:

LOAD n2_flp1_fred carga el fichero "fred" del disco 1 de la estación 2

OPEN_IN #3,n1_flp2_miprograma abre "miprograma" del disco 2 de la estación 1

OPEN #3,n1_con_120x20a0x0 abre una ventana de 20 columnas por 2 filas en la estación 1

El uso de nombres de directorio de omisión hace este trabajo bastante más fácil. Por ejemplo:

PROG_USE n1_win1_progs por omisión, todos los programas serán cargados desde el directorio "progs" del disco winchester 1 de la estación 1

SPL_USE n1_ser selecciona SER1 de la estación 1 como destino del "spooler" de omisión

Es posible ocultar la red local a las aplicaciones escogiendo un nombre especial para el servidor de ficheros en red local.

NFS_USE *nombre,nombres de red local* selecciona el fichero de red local

Los "nombres de red local" deben ser nombres de directorio completos, pudiéndose dar hasta 8 en un comando. Cada uno de estos nombres estará asociado con uno de los 8 posibles dispositivos de directorio ("nombre"1 a "nombre"8).

Por ejemplo

NFS_USE mdv,n2_flp1_,n2_flp2_ selecciona nombre del servidor de ficheros en red local de manera que cualquier referencia

a "mdv1" en este QL remoto, será tomada como una llamada a flp1_ en la estación 2, mientras que "mdv2_" será tomado como flp1_ en estación 2

OPEN_NEW #3,mdv2_fred ahora este comando abrirá el fichero "fred" en el disco 2 de la estación 2

Los nombres de red local serán normalmente un número de la red local seguido de un nombre de dispositivo, como anteriormente, debiendo terminar con un subrayado (_) para indicar que el nombre es un directorio. Por suspuesto, si el nombre del servidor de ficheros en red local se usa junto con los comandos "wild card" de mantenimiento de ficheros, ésta es la única forma aceptable. QUILL, sin embargo, tiende a abrir un fichero con el nombre DEF_TMP en mdv2_. Puede haber problemas si se ejecuta más de una copia de QUILL a través de la red local al mismo tiempo. Ello se puede evitar si se selecciona un directorio para el nombre de red local de mdv2_:

NFS_USE mdv,n1_flp1_,n1_flp2_fred_ Ahora, DEF_TMP abierto en mdv2_ aparecerá en el directorio "fred" del flp2_ de la estación 1

22.4 Mensajería

Las facilidades de red local del Toolkit II pueden ser usadas también para mensajerías. Puede abrirse una ventana, enviar un mensaje, y leer una replica usando un simple programa en SuperBASIC. Si se necesita un mensaje particularmente bonito, se pueden utilizar las capacidades gráficas del SuperBASIC. Las únicas facilidades de entrada/salida estándard no disponibles a través de la red local son SD.EXTOP (operaciones extendidas) y SD.FOUNT (elegir fuentes).

Por ejemplo:

```
ch=FOPEN (n2_con_150x10a0x0):CLS #ch
INPUT #ch,"¿Quieres un café?";rep$
IF "y" INSTR rep$=1: PRINT "Fred quiere un café"
CLS #ch:CLOSE #ch
```

---

23 Escribiendo Programas para usar con EX

Los programas invocados mediante **EX** (o **EW** o **ET**) pueden ser de tres tipos:

no estándares la cabecera del programa no tiene un formato estándar

especial la cabecera del programa es estándard pero tiene una señal adicional

estándard la cabecera del programa es estándard

En lo que respecta a **EX**, la distinción estriba en que los programas especiales deben contener el código para abrir sus propios canales de entrada/salida.

23.1 Programas Estándar y No Estándard

Al comienzo de la ejecución de un programa estándard o no estándard, debe haber la suguiente información en la pila:

-- El número total de canales abiertos por la tarea (palabra)
-- [La identificación del canal del "pipe" de entrada, si presente (palabra larga)]
-- La identificación del canal de cada nombre de fichero dado en la especificación del programa (palabra larga)
-- [La identificación del canal del "pipe" de salida, si presente (palabra larga)]
-- La longitud de la cadena opcional o un cero (palabra)
-- [La cadena opcional (bytes)]

Si sólo hay un canal abierto para una tarea, estará abierto para lectura/escritura a no ser que sea un "pipe" en cuyo caso la dirección estará implicada en el comando.

Si hay más de un canal abierto para una tarea, el primer canal será la entrada principal (abierto sólo para lectura), y el resto será abierto con sobreescritura (OVERWRITE). El último canal es la salida principal.

Una tarea no debe cerrar los canales suministrados, sino que debe suicidarse cuando esté completa. Cada tarea es propiedad de la siguiente de la cadena, de forma que cuando la última tarea acaba, toda la cadena es eliminada. Cuando se suicide la cadena de esta manera, se colocará una señal "fin de fichero" en la salida. Una señal "fin de fichero" en su canal principal de entrada indicará, directamente o de otra manera, a una programa que los datos están completos.

23.2 Programas Especiales

Los programas estándard y los especiales contienen el valor 4AFB en sus bytes 6 y 7. A estos bytes le sigue la identificación del programa en forma de cadena estándard (una palabra con su longitud seguida de los bytes de los caracteres de la cadena). En el caso del encabezamiento de programas especiales, la identifiación es seguida por otro valor $4AFB (alineado con los límites de una palabra). Una vez cargado el programa, puesta la cadena opcional en la pila de las tareas y abierto el "pipe" de entrada (en los casos necesarios) y puesta su identificación en la pila de la tarea, **EX** hará una llamada a la dirección señalada después de la segunda palabra de identificación. Obsérvese que el código llamado deberá formar parte de una procedimiento del Basic y no de un programa ejecutable.

Al entrar en este código, los siguientes registros deben ser seleccionados a los siguientes valores:

D4.L 0 or 1 si hay un "pipe" de entrada: ID no está en la pila
D5.L 0 o 1 si hay un "pipe" de salida: ID no está en la pila
D6.L ID de Tarea para este programa
D7.L número total de "pipes" + el de nombres de fichero en la especificación del programa

A0 dirección de las rutinas de apoyo
A1 puntero a la cadena de comando
A3,A6 (*)puntero al primer nombre de fichero (tabla de nombres)
A4 puntero a la pila de la tarea
A5,A6 (*)puntero por debajo del último nombre de fichero (tabla de nombres)
(*) estos son los registros estándard para el paso de parámetros de procedimientos del Basic.
El procedimiento de selección de fichero decodifica los nombres de los ficheros, los abre y pone su identificación (ID) en la pila (A4). El registro D0 devolverá el código de error. D5 será aumentado con el número de identificaciones de canal puestos en la pila de la tarea. En A4 se mantendrá el puntero a la pila de la tarea. La información contenida en los registros D1 a D7, A0 a A3 y A5 será considerada como volátil.
La rutina (A0) para obtener un nombre de fichero deberá ser llamada con el puntero en A3 señalando a la entrada apropiada en la tabla de nombres. D0 devuelve el código de error. D1 a D3 son destruídos. Si D0 es igual a 0, A1 devolverá el puntero al nombre (relativo a A6). Si D0 es positivo, A0 devolverá la identificación del canal del SuperBasic (si el parámetro era #n), siendo preservados todas los demás registros de direcciones.
La rutina 2(A0) para abrir un canal deberá ser llamada con el puntero al nombre del fichero en A1(relativa a A6). El nombre del fichero no debe estar en el amortiguador del Basic; D3 debe contener el código de acceso (es posible la sobreescritura) y la identificación de la tarea (tal como es pasada por la rutina de inicialización) deberá estar en D6. El código de error será devuelto en D0 mientras que D1 y D2 serán destruídos. A1 será devuelto apuntando al nombre del fichero usado (puede tener delante un directorio de omisión). Si la apertura falla, A1 apuntará al nombre del fichero de omisión + el dado. La identificación del canal es devuelta en A0 y todos los demás registros son preservados.
En ambos casos, el registro de estado es devuelto seleccionado de acuerdo con el valor de D0.

Apéndice A

Lista de Diferencias
Este índice lista las extensiones al SuperBasic en orden alfabético junto con su tipo (procedimiento, función, programa), el número de la sección en la que es descrita detalladamente, su origen (en el caso de que apareciera ya en las ROMs del QL o en el Sinclair QL Toolkit) y las

principales diferencias entre la actual y las anteriores versiones.
La lista sólo incluye las diferencias más importantes, aunque en algunos casos hay otras importantes mejoras con respecto a las versiones anteriores.

Nombre	***Tipo***	***Sec.***	***Origen***	***Diferencias***
AJOB	proced.	9	QL Toolkit	acepta nombre tarea
ALARM	programa	18	QL Toolkit	programa residente
ALCHP	función	15	QL Toolkit	
ALTKEY	proced.	20	nuevo	
BGET	proced.	12	QL Toolkit	
BIN	función	13	QL Toolkit	
BIN$	función	13	QL Toolkit	
BPUT	proced.	12	QL Toolkit	
CALL	proced.	7	QL	error eliminado
CDEC$	función	13	QL Toolkit	
CHAR_USE	proced.	14	QL Toolkit	
CHAR_INC	proced.	14	QL Toolkit	
CLCHP	proced.	15	QL Toolkit	
CLEAR	proced.	6	QL	limpia WHEN ERROR
CLOCK	programa	18	QL Toolkit	programa configurable
CLOSE	proced.	10	QL	cierra varios ficheros
CONTINUE	proced.	17	QL	admite número de línea
COPY	proced.	5	QL	usa directorio omisión
				usa destino omisión
COPY_O	proced.	5	nuevo	sobreescribe fichero
COPY_N	proced.	5	QL	usa directorio omisión
				usa destino omisión
COPY_H	proced.	5	nuevo	
CURSEN	proced.	14	QL Toolkit	
CURDIS	proced.	14	QL Toolkit	
DATA_USE	proced.	4	QL Toolkit	
DATAD$	función	4	nuevo	
DDOWN	proced.	4	nuevo	
DEL_DEFB	proced.	15	nuevo	
DELETE	proced.	5	QL	usa directorio omisión
DEST_USE	proced.	4	nuevo	
DESTD$	función	4	nuevo	
DIR	proced.	5	QL	usa directorio omisión
DLIST	proced.	4	nuevo	
DO	proced.	6	nuevo	
DNEXT	proced.	4	nuevo	
DUP	proced.	4	nuevo	
ED	proced.	3	QL Toolkit	completamente cambiado
ERR_DF	función	17	QL	error eliminado
ET	proced.	8	QL Toolkit	
EX	proced.	8	QL Toolkit	

EXEC	proced.	8	QL	ahora igual a EX
EXEC_W	proced.	8	QL	ahora igual a EW
EXTRAS	proced.	19	QL Toolkit	
EW	proced.	8	QL Toolkit	
FDAT	función	11	QL Toolkit	
FDEC$	función	13	QL Toolkit	
FEXP$	función	13	nuevo	
FLEN	función	11	QL Toolkit	
FLUSH	proced.	12	nuevo	
FNAME$	función	11	nuevo	
FOP_DIR	función	10	QL Toolkit	encuentra canal libre
FOP_IN	función	10	QL Toolkit	encuentra canal libre
FOP_NEW	función	10	QL Toolkit	encuentra canal libre
FOP_OVER	función	10	QL Toolkit	encuentra canal libre
FOPEN	función	10	QL Toolkit	encuentra canal libre
FPOS	función	12	QL Toolkit	
FREE_MEM	función	15	QL Toolkit	da 512 bytes menos
FSERVE	programa	22	nuevo	
FTEST	función	10	nuevo	
FTYP	función	11	QL Toolkit	
FUPDT	función	11	nuevo	
FXTRA	función	11	nuevo	
GET	proced.	12	QL Toolkit	
HEX	función	13	QL Toolkit	
HEX$	función	13	QL Toolkit	
IDEC$	función	13	QL Toolkit	
JOB	proced.	9	QL Toolkit	
JOB$	función	9	QL Toolkit	
LBYTES	proced.	7	QL	usa directorio omisión
LOAD	proced.	6	QL	usa directorio omisión limpia WHEN ERROR
LRESPR	proced.	7	nuevo	
LRUN	proced.	6	QL	usa directorio omisión limpia WHEN ERROR
MERGE	proced.	6	QL	usa directorio omisión limpia WHEN ERROR
MRUN	proced.	6	QL	usa directorio omisión limpia WHEN ERROR
NEW	proced.	6	QL	limpia WHEN ERROR
NFS_USE	proced.	22	nuevo	
NXJOB	función	9	QL Toolkit	
OJOB	función	9	QL Toolkit	
OPEN	proced.	10	QL	usa directorio omisión
OPEN_DIR	proced.	10	nuevo	usa directorio omisión
OPEN_IN	proced.	10	QL	usa directorio omisión
OPEN_NEW	proced.	10	QL	usa directorio omisión
OPEN_OVER	proced.	10	nuevo	usa directorio omisión

PARNAM$ función 16 nuevo
PARSTR$ función 16 nuevo
PARTYP función 16 QL Toolkit
PARUSE función 16 QL Toolkit
PJOB función 9 QL Toolkit
PRINT_USING proced. 13 nuevo
PROG_USE proced. 3 QL Toolkit
PROGD$ función 3 nuevo
PUT proced. 12 QL Toolkit
RECHP proced. 15 QL Toolkit
RENAME proced. 5 QL Toolkit
RETRY proced. 17 QL admite número de línea
RJOB proced. 9 QL Toolkit acepta nombre de tarea
RUN proced. 6 QL limpia WHEN ERROR
SAVE proced. 6 QL usa directorio omisión
SAVE_O proced. 6 nuevo sobregraba fichero
SBYTES proced. 7 QL usa directorio omisión
SBYTES_O proced. 7 nuevo sobregraba fichero
SEXEC proced. 7 QL usa directorio omisión
SEXEC_O proced. 7 nuevo sobregraba fichero
SPJOB proced. 9 QL Toolkit acepta nombre de tarea
SPL programa 5 QL Toolkit destino simplificado
SPL_USE proced. 4 QL Toolkit
SPLF programa 5 nuevo añade salto de página
al fichero
STAT proced. 5 QL Toolkit
STOP proced. 6 QL limpia WHEN ERROR
TK2_EXT proced. 20 nuevo
TRUNCATE proced. 12 QL Toolkit la posición puede ser
especificada
VIEW proced. 3 QL Toolkit la ventana de comandos
es el canal de omisión
WCOPY proced. 5 nuevo la ventana de comandos
es el canal de omisión
usa destino de omisión
WDEL proced. 5 QL Toolkit
WDIR proced. 5 QL Toolkit
WMON proced. 14 QL Toolkit
WREN proced. 5 nuevo la ventana de comandos
es el canal de omisión, usa
destino de omisión
WTV proced. 14 QL Toolkit
WSTAT proced. 5 QL Toolkit

Apéndice B

Este apéndice ilustra el uso de las facilidades del Toolkit II con el ensamblador y "linkador" de GST (la versión usada por QJUMP es la suministrada por GST con su compilador QC: ¡ merece la pena comprar QC incluso sólo por tener el ensamblador y el "linkador"!).
Los programas aceptan una gran variedad de opciones en su línea de comandos. Esta línea de comandos puede ser pasada a los programas en el parámetro-cadena del comando **EX.** Desafortunadamente, los programas no tratan de encontrar el directorio de omisión para datos, por lo que es necesario añadirlo a los nombres de los ficheros en las línea de comandos.
El ensamblador se llama **ASM** y el "linkador" **LINK.** Los nombres de ficheros pueden ser pasados a estos procedimientos como cadenas o como nombres.

```
100 REMark ensamblado de un fichero relocalizable
110 :
120 DEFine PROCedure asr(file$)
130 EX asm;DATAD$&PARSTR$(file$,1)&"-errors scr"
140 END DEFine asr
150 :
160 REMark ensamblado con listado
170 :
180 DEFine PROCedure asl(file$)
190 EW asm;DATAD$&PARSTR$(file$,1)&"-list ser - nosym"
200 END DEFine asl
210 :
220 REMark programa de "linkado"
230 :
240 DEFine PROCedure lk(file$)
250 EX link;DATAD$&PARSTR$(file$,1)&"-with"&DATAD$&"link -nolist"
260 END DEFine lk
```

Si el directorio de omisión es "**FLP1_JUNK_**", los procedimientos *ASL "table"* y *LK master* crearán las cadenas de parámetros de comandos en el ensamblador y "linkador"
"FLP1_JUNK_table -list ser -nosym"
y
"FLP1_JUNK_master -with FLP1_JUNK_link -nolist"

Apéndice C

Protocolos de Red Local del QL

Concordancia estándard QL

La concordancia estándard QL de protocolo de red local es compatible con el protocolo del Sinclair Spectrum. Comprende 11 fases.

	emisor	**receptor**
a)scout		
1)gap	esperando 3 ms para actividad, si ocurre actividad, recomenzar	
2)wait		esperando para actividad (una exploración)
3)scout	enviar una exploración de duración <530us, si ocurre contestación: recomenzar	esperar 530 us
b)header		
4)hactiv	elegir net activa 22 us	esperar activa
5)hbytes	para cada byte 11.2us bit comienzo (inactiva),bits de datos 8*11.2us, 5*11.2us bits de parada(activa)	para cada bit esperar bit comienzo (inactiva),leer 8 bits de datos, si falla:recomenzar
6)hackw	esperar 2.5ms para activa, si no activa:recomenzar	seleccionar net activa 22us
7)hackbt	esperar bit comienzo,leer ocho bits de datos, si error:recomenzar	enviar 11.2us bit comienzo, 8 bits de datos,00000001
c)data		
8)dactiv	seleccionar net activa 22 us	esperar para activa
9)dbytes	para cada byte 11.2us bit comienzo (inactiva),bits de datos 8*11.2us, 5*11.2us bits de parada(activa)	para cada bit esperar bit comienzo (inactiva),leer 8 bits de datos, si falla:recomenzar
10)dackw	esperar 2.5ms para activa, si no activa:recomenzar	seleccionar net activa 22 us
11)dackbt	esperar para bit de comienzo, leer 8 datos	enviar 11.2us bit de comienzo, 8 bits de datos, bits 00000001, si error:recomenzar

scout=exploración; gap= espacio; wait=espera; header=cabecera; Not acknowledge= no reconocida;

El protocolo entero es sincronizado mediante un periodo de inactividad de al menos 2.8 ms de longitud.
La cabecera tiene una longitud de ocho bytes en el siguiente formato:
número de estación de destinación
número de estación de envio
número de bloque (byte alto)
número de bloque (byte bajo)
tipo de bloque (0=normal, 1=último bloque del fichero)
número de bytes en el bloque (de 0 a 255)
suma de comprobación de los datos
suma de comprobación de la cabecera
Si el número de bytes en el bloque es 0, quiere decir que se están enviando 256 bytes.
Las sumas de comprobación se forman por simple adición: si hay dos errores en un mismo bloque en dos bits individuales en el bit más significativo (el tipo de error más común), el error pasará indetectado.
Si el número de bloque recibido en una cabecera no es igual al número de bloque requerido, la cabecera y los datos son "reconocidos" pero ignorados.
El protocolo no está protegido contra un fallo en el último bloque transmitido en el que el receptor ha aceptado el bloque pero el emisor ha perdido el "reconocimiento". En ese caso, el emisor seguirá manteniendo la retransmisíón hasta que sa acabe el tiempo (20 segundos).

Emisión en el Toolkit II
El toolkit II tiene una versión especial de este protocolo para la emisión en la red local. Tiene una exploración prolongada para permitir que el receptor interroge al IPC sin perder la exploración, y tiene un reconocimiento/no reconocimiento activo. El protocolo ha sido definido de tal manera que permitirá que los controladores de red local futuros puedan ser más flexibles que el actual.

	emisor	**receptor**
a)scout		
1)gap	esperando 3 ms para actividad, si no ocurre actividad, recomenzar	
2)wait		esperando para actividad (una exploración), revisión del IPC cada 20 ms por BREAK
3)scout	enviar una explora-	esperar 530 us

ción de duración
<530us, si ocurre
contestación:
recomenzar

4)scext enviar una extensión de exploración de 5ms activa

b)header

5)hbytes para cada byte para cada bit esperar
11.2us bit comienzo bit comienzo
(inactiva),bits de (inactiva),leer 8
datos 8*11.2us, bits de datos, si
5*11.2us bits de falla:nack
parada(activa)

6)hwait dejar net activa,
esperar un segundo

c)data

7)dbytes para cada byte para cada bit esperar
11.2us bit comienzo bit comienzo
(inactiva),bits de (inactiva),leer 8
datos 8*11.2us, bits de datos, si
5*11.2us bits de falla:nack
parada(activa)

8)dack net inactiva y durante 500us,selec- esperar 1ms
para cionar net activa y
activa:si falla: esperar 5ms,hacer
recomenzar cualquier proceso
requerido y cuando
listo para próximo
paquete,inactivar y
recomenzar

d)Not acknowledge

9)nack esperar para esperar 2.8us de inactiva
activa o inactiva,
si inactiva, recomenzar

10)nackw esperar 500us para esperar 200us para
activa:fuera de activa,si activa,
tiempo es correcto, recomenzar,si inac-
activa es fallo tiva, activar 500us
(nack)

scout=exploración; gap= espacio; wait=espera; header=cabecera; Not acknowledge= no reconocida;

Una emisión reconocida es activa 5 ms seguida de 400 us inactiva. Una emisión no reconocida no responde o es activa 5 ms seguida por 200 a 300 us inactiva, seguida de más de 200 us activa.

Protocolo del servidor en el toolkit II

El protocolo del servidor en el toolkit II es físicamente el mismo que

el estándard QL, pero la cabecera ha sido ligeramente cambiada para mejorar la suma de comprobación, para permitir el envio de bloques de hasta 1000 bytes, y para distinguir las transacciones del servidor. Una cabecera del servidor no puede ser confundida con una cabecera estándard.

espera durante 500 us para activa: fuera de tiempo es correcta, activa es fallo	espera 200 us para activa. si activa: recomienza, si inactiva, activar 500 us (nack)

PROGRAMAS DE ESTE NUMERO

- TRANSBASE_EXE.- La versión EXEC para todo el mundo.
- ENC50_BAS.- Un programa de encuestas alimentarias para médicos escrito por Miguel Frasquet.

EN PROXIMOS NUMEROS...

El manual del Qliberator traducido.

Hay muchos manuales traducidos en stock, pero casi ninguna colaboración dedicada a otras secciones.